



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/717,675	11/21/2000	Christopher G. Kaler	777.340US1	9107

23377 7590 05/07/2004
WOODCOCK WASHBURN LLP
ONE LIBERTY PLACE, 46TH FLOOR
1650 MARKET STREET
PHILADELPHIA, PA 19103

EXAMINER

ALI, SYED J

ART UNIT PAPER NUMBER

2127

DATE MAILED: 05/07/2004

5

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary

Application No.

09/717,675

Applicant(s)

KALER ET AL

Examiner

Syed J Ali

Art Unit

2127

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) ☒ Responsive to communication(s) filed on 21 November 2000.
- 2a) ☐ This action is **FINAL**. 2b) ☒ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) ☒ Claim(s) 1-71 is/are pending in the application.
- 4a) Of the above claim(s) _____ is/are withdrawn from consideration.
- 5) ☐ Claim(s) _____ is/are allowed.
- 6) ☒ Claim(s) 1-71 is/are rejected.
- 7) ☐ Claim(s) _____ is/are objected to.
- 8) ☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☐ The drawing(s) filed on _____ is/are: a) ☐ accepted or b) ☐ objected to by the Examiner.
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some * c) ☐ None of:
1. ☐ Certified copies of the priority documents have been received.
 2. ☐ Certified copies of the priority documents have been received in Application No. _____.
 3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

- | | |
|--|---|
| 1) <input checked="" type="checkbox"/> Notice of References Cited (PTO-892) | 4) <input type="checkbox"/> Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____ |
| 2) <input checked="" type="checkbox"/> Notice of Draftsperson's Patent Drawing Review (PTO-948) | 5) <input type="checkbox"/> Notice of Informal Patent Application (PTO-152) |
| 3) <input type="checkbox"/> Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08)
Paper No(s)/Mail Date _____ | 6) <input type="checkbox"/> Other: _____ |

DETAILED ACTION

1. Claims 1-71 are pending in this application.

Claim Objections

2. Claim 65 is objected to because of the following informalities:
 - a. In line 2 of claim 65, "a computer system for" should read "a computer system".Appropriate correction is required.

Claim Rejections - 35 USC § 112

3. The following is a quotation of the second paragraph of 35 U.S.C. 112:

The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

4. Claim 40 is rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

5. The following claim language is indefinite:

- a. Claim 40 recites, "the pool of threads includes a number of threads in a range from about $N+1$ to about $2*N$ ". The use of the word "about" leaves the claimed number of threads in the thread pool vague, and thus the scope of the claim is unascertainable.

Claim Rejections - 35 USC § 102

6. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

Art Unit: 2127

A person shall be entitled to a patent unless –

(e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

7. Claims 41-48 are rejected under 35 U.S.C. 102(e) as being anticipated by Ramakrishnan et al. (USPN 6,085,215) (hereinafter Ramakrishnan).

8. As per claim 41, Ramakrishnan teaches the invention as claimed, including a method in a computer system for servicing requests from multiple client computers, the method comprising:

monitoring a quantity of work being performed by the computer system (col. 8 lines 13-39; col. 11 line 52 - col. 12 line 8);

determining whether the quantity has exceeded an upper limit (col. 8 lines 13-39; col. 11 line 52 - col. 12 line 8); and

if the quantity has exceeded the upper limit but has not dropped below a lower limit, not accepting new requests into the computer system (col. 8 lines 13-39; col. 11 line 52 - col. 12 line 8).

9. As per claim 42, Ramakrishnan teaches the invention as claimed, including the method as claimed in claim 41, further comprising, if the quantity has exceeded the upper limit and has dropped below the lower limit, accepting the new requests into the computer system (col. 8 lines 13-39; col. 11 line 52 - col. 12 line 8).

Art Unit: 2127

10. As per claim 43, Ramakrishnan teaches the invention as claimed, including the method as claimed in claim 41, further comprising continuing to monitor the quantity of work being performed by the computer system after the quantity has exceeded the upper limit (col. 8 lines 13-39; col. 11 line 52 - col. 12 line 8).

11. As per claim 44, Ramakrishnan teaches the invention as claimed, including the method as claimed in claim 41, wherein the quantity of work is indicated by a number of work items on one or more work queues, and determining whether the quantity has exceeded the upper limit comprises determining whether the number of work items has exceeded a predefined value (col. 8 lines 13-39; col. 11 line 52 - col. 12 line 8).

12. As per claim 45, Ramakrishnan teaches the invention as claimed, including a method in a computer system for servicing requests from multiple client computers, the method comprising:

monitoring an amount of time to return a result by the computer system (col. 8 lines 13-39; col. 11 line 52 - col. 12 line 8);

determining whether the amount of time has exceeded an upper limit (col. 8 lines 13-39; col. 11 line 52 - col. 12 line 8); and

if the amount of time has exceeded the upper limit but has not dropped below a lower limit, not processing new work items (col. 8 lines 13-39; col. 11 line 52 - col. 12 line 8).

13. As per claim 46, Ramakrishnan teaches the invention as claimed, including the method as claimed in claim 45, further comprising, if the amount of time has exceeded the upper limit and

Art Unit: 2127

has dropped below the lower limit, accepting and processing the new work items (col. 8 lines 13-39; col. 11 line 52 - col. 12 line 8).

14. As per claim 47, Ramakrishnan teaches the invention as claimed, including the method as claimed in claim 45, further comprising continuing to monitor the amount of time after the amount of time has exceeded the upper limit (col. 8 lines 13-39; col. 11 line 52 - col. 12 line 8).

15. As per claim 48, Ramakrishnan teaches the invention as claimed, including the method as claimed in claim 45, wherein the amount of time includes a time it takes to post the result to an output port, and determining whether the amount of time has exceeded the upper limit comprises determining whether the time it takes to post the result has exceeded a predefined value (col. 8 lines 13-39; col. 11 line 52 - col. 12 line 8).

Claim Rejections - 35 USC § 103

16. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

17. **Claims 1-8, 38-40, 49-52, 55-56, 58-59, 65-66 are rejected under 35 U.S.C. 103(a) as being unpatentable over Huff et al. (USPN 6,457,064) (hereinafter Huff) in view of Hanson et al. (USPN 6,697,835).**

18. As per claim 1, Huff teaches the invention substantially as claimed, including a method in a computer system for servicing requests from one or more client computers, the method comprising:

receiving a request from a client computer (col. 4 lines 40-58; col. 6 lines 12-55);

a first thread processing the request by invoking a receive handler that creates a work item, wherein the first thread is part of a pool of generic threads (col. 4 lines 40-58; col. 6 lines 12-55); and

a second thread performing a task specified in the work item by invoking a work handler, wherein the second thread is part of the pool of generic threads (col. 6 lines 12-55; col. 7 lines 17-38).

19. Hanson teaches the invention as claimed, including the following limitations not shown by Huff:

receiving a result of performing the task (col. 6 lines 9-54); and

a third thread returning at least a portion of the result to the client computer by invoking a reply handler, wherein the third thread is part of the pool of generic threads (col. 6 lines 9-54).

20. It would have been obvious to one of ordinary skill in the art to combine Huff and Hanson since the thread allocation procedure taught by Huff deals only with allocating an incoming request to a thread for processing, but fails to specifically state how the processing results are returned to the client. Hanson makes up for this deficiency with a method of invoking a reply thread in a distributed processing system that allows the agent or server to handle all

Art Unit: 2127

requests transparently from the client. This simplifies the processing and allows system threads to be allocated in an efficient manner.

21. As per claim 2, Huff teaches the invention as claimed, including the method as claimed in claim 1, wherein receiving the request comprises:

receiving the request into an input/output port (col. 4 lines 40-58); and

placing a reference in a queue indicating that work is available for the first thread (col. 6 lines 12-45).

22. As per claim 3, Huff teaches the invention as claimed, including the method as claimed in claim 2, further comprising:

invoking a receive thread manager when the work is available (col. 6 lines 12-40); and

the receive thread manager scheduling the first thread for execution on one of multiple processors (col. 6 lines 12-40; col. 9 lines 19-34).

23. As per claim 4, Huff teaches the invention as claimed, including the method as claimed in claim 3, wherein the receive thread manager schedules the first thread from a queue of available threads (col. 6 lines 12-40).

24. As per claim 5, Huff teaches the invention as claimed, including the method as claimed in claim 1, further comprising the first thread placing the work item on a work queue for execution by the second thread (col. 6 lines 12-40).

25. As per claim 6, Huff teaches the invention as claimed, including the method as claimed in claim 5, further comprising the first thread placing a reference in a completion port queue, indicating that work is available for the second thread (col. 6 lines 12-40).

26. As per claim 7, Hanson teaches the invention as claimed, including the method as claimed in claim 1, further comprising the second thread creating and placing a second work item on a reply work queue when the results are received (col. 6 lines 27-40).

27. As per claim 8, Hanson teaches the invention as claimed, including the method as claimed in claim 7, further comprising the second thread placing a reference in a completion port queue, indicating that work is available for the third thread (col. 6 lines 27-40).

28. As per claim 38, Huff teaches the invention as claimed, including a method in a computer system for servicing requests from one or more client computers, the method comprising:

maintaining a pool of threads, wherein each thread in the pool of threads is identical and can invoke at least one receive handler and at least one work handler (col. 4 lines 40-58; col. 6 lines 12-55);

invoking receive handlers by the threads in response to receiving requests from one or more client computers, wherein the receive handlers create work items that specify tasks to be performed to satisfy the request (col. 4 lines 40-58; col. 6 lines 12-55); and

Art Unit: 2127

invoking work handlers by the threads to perform the tasks specified in the work items (col. 6 lines 12-55; col. 7 lines 17-38).

29. Hanson teaches the invention as claimed, including the following limitations not shown by Huff:

maintaining a pool of threads, wherein each thread in the pool of threads is identical and can invoke at least one reply handler (col. 6 lines 9-54);

receiving results of the tasks (col. 6 lines 9-54); and

invoking reply handlers by the threads to return at least portions of the results to the client computers (col. 6 lines 9-54).

30. As per claim 39, Huff teaches the invention as claimed, including the method as claimed in claim 38, wherein one or more of the receive handlers, one or more of the work handlers, and one or more of the reply handlers can be simultaneously executed on multiple processors available to the computer system (col. 6 lines 12-40; col. 9 lines 19-34).

31. As per claim 40, Huff teaches the invention as claimed, including the method as claimed in claim 38, wherein the pool of threads includes a number of threads in a range from about $N+1$ to about $2*N$, where N is a number of processors available to the computer system (col. 4 lines 40-58; col. 6 lines 12-55).

32. As per claim 49, Huff teaches the invention as claimed, including an application program for implementation by an application server, the application program comprising:

at least one receiver handler that can be invoked by a thread within a pool of threads (col. 4 lines 40-58; col. 6 lines 12-55); and

at least one work handler that also can be invoked by the thread (col. 6 lines 12-55; col. 7 lines 17-38).

33. Hanson teaches the invention as claimed, including the following limitations not shown by Huff:

at least one reply handler that also can be invoked by the thread (col. 6 lines 9-54).

34. As per claim 50, Huff teaches the invention as claimed, including the application program as claimed in claim 49, wherein a receiver handler of the at least one receiver handler is executed when the application server receives a request from a client computer, and the receiver handler creates a first work item to be performed by a work handler of the at least one work handler (col. 4 lines 40-58; col. 6 lines 12-55).

35. As per claim 51, Huff teaches the invention as claimed, including the application program as claimed in claim 50, wherein the work handler is executed when the first work item exists, the work handler receives results (col. 6 lines 12-55; col. 7 lines 17-38).

36. Hanson teaches the invention as claimed, including the following limitations not shown by Huff:

the work handler creates a second work item to be performed by a reply handler of the at least one reply handler (col. 6 lines 9-54).

Art Unit: 2127

37. As per claim 52, Hanson teaches the invention as claimed, including the application program as claimed in claim 51, wherein the reply handler is executed when the second work item exists, and the reply handler sends the results to the client computer (col. 6 lines 9-54).

38. As per claim 55, Huff teaches the invention as claimed, including the application program as claimed in claim 49, wherein multiple identical copies of the thread exist within the pool of threads, and at least one copy of the thread is executed each time a request is received from a client computer (col. 4 lines 9-29; col. 6 lines 12-55).

39. As per claim 56, Huff teaches the invention as claimed, including the application program as claimed in claim 49, wherein multiple copies of the thread can simultaneously be executed by multiple processors available to the application server (col. 6 lines 12-40; col. 9 lines 19-34).

40. As per claim 58, Huff teaches the invention as claimed, including a computer system for servicing requests from one or more client computers, the computer system comprising:

a memory containing an application server which maintains a pool of threads, wherein each thread in the pool of threads is identical and can invoke at least one receive handler and at least one work handler (col. 4 lines 40-58; col. 6 lines 12-55; col. 7 lines 17-38), and the application server further schedules threads in the pool of threads for execution on multiple processors available to the computer system (col. 6 lines 12-40; col. 9 lines 19-34); and

the multiple processors for simultaneously executing multiple threads within the pool of threads (col. 6 lines 12-40; col. 9 lines 19-34).

41. Hanson teaches the invention as claimed, including the following limitations not shown by Huff:

each thread in the pool of threads is identical and can invoke at least one reply handler (col. 6 lines 9-54).

42. As per claim 59, Huff teaches the invention as claimed, including the computer system as claimed in claim 58, wherein the application server further performs functions of:

receiving a request from a client computer (col. 4 lines 40-58; col. 6 lines 12-55);

a first thread processing the request by invoking a receive handler which creates a work item, wherein the first thread is part of the pool of threads (col. 4 lines 40-58; col. 6 lines 12-55);
and

a second thread performing a task specified in the work item by invoking a work handler, wherein the second thread is part of the pool of threads (col. 6 lines 12-55; col. 7 lines 17-38).

43. Hanson teaches the invention as claimed, including the following limitations not shown by Huff:

receiving a result of performing the task (col. 6 lines 9-54); and

a third thread returning at least a portion of the result to the client computer by invoking a reply handler, wherein the third thread is part of the pool of threads (col. 6 lines 9-54).

Art Unit: 2127

44. As per claim 65, Huff teaches the invention as claimed, including a computer-readable medium holding computer executable instructions, the computer-readable medium for performing a method in a computer system, the method comprising:

maintaining a pool of threads, wherein each thread in the pool of threads is identical and can invoke at least one receive handler and at least one work handler (col. 4 lines 40-58; col. 6 lines 12-55; col. 7 lines 17-38), and the application server further schedules threads in the pool of threads for execution on multiple processors available to the computer system (col. 6 lines 12-40; col. 9 lines 19-34); and

simultaneously executing multiple threads within the pool of threads on multiple processors available to the computer system (col. 6 lines 12-40; col. 9 lines 19-34).

45. Hanson teaches the invention as claimed, including the following limitations not shown by Huff:

each thread in the pool of threads is identical and can invoke at least one reply handler (col. 6 lines 9-54).

46. As per claim 66, Huff teaches the invention as claimed, including the computer-readable medium as claimed in claim 65, wherein the method further comprises:

receiving a request from a client computer (col. 4 lines 40-58; col. 6 lines 12-55);

a first thread processing the request by invoking a receive handler, which creates a work item, wherein the first thread is part of the pool of threads (col. 4 lines 40-58; col. 6 lines 12-55);
and

a second thread performing a task specified in the work item by invoking a work handler, wherein the second thread is part of the pool of threads (col. 6 lines 12-55; col. 7 lines 17-38).

47. Hanson teaches the invention as claimed, including the following limitations not shown by Huff:

receiving a result of performing the task (col. 6 lines 9-54); and

a third thread returning at least a portion of the result to the client computer by invoking a reply handler, wherein the third thread is part of the pool of threads (col. 6 lines 9-54).

48. **Claims 9-10 are rejected under 35 U.S.C. 103(a) as being unpatentable over Huff in view of Hanson as applied to claim 1 above, and further in view of Singh (USPN 6,665,704).**

49. As per claim 9, Huff teaches the invention as claimed, including the method as claimed in claim 1, further comprising:

receiving input data (col. 5 line 47 - col. 6 line 11).

50. Singh teaches the invention as claimed, including the following limitations not shown by Huff:

the first thread storing the input data in a cache that is accessible to the second thread (col. 1 line 62 - col. 2 line 45).

51. It would have been obvious to combine the modified Huff and Singh since the methods of both Huff and Hanson refer to cache coherency, but fail to specifically state how it is maintained between threads. Singh provides a way of loading data in a cache for a second thread to perform further processing. This greatly improves the processing speed of the overall system,

Art Unit: 2127

as the time to access data in cache is much faster than performing retrieval from main memory. This improves the server's response time and provides results to the client in a much quicker manner.

52. As per claim 10, Singh teaches the invention as claimed, including the method as claimed in claim 1, wherein the results include data, and further comprising storing the data in a cache that is accessible to the third thread (col. 1 line 62 - col. 2 line 45).

53. **Claims 11-12 are rejected under 35 U.S.C. 103(a) as being unpatentable over Huff in view of Hanson as applied to claim 1 above, and further in view of Fant (USPN 6,327,607).**

54. As per claim 11, Fant teaches the invention as claimed, including the method as claimed in claim 1, further comprising:

determining a size of the result (col. 4 lines 11-56); and

when the size of the result is not larger than a cutoff size, storing the result in the partial results cache (col. 4 lines 11-56).

55. It would have been obvious to one of ordinary skill in the art to combine the modified Huff and Fant since the use of a partial cache to store incremental results ensures that the results are not lost due to overflow errors, while also providing the return threads with a data store that is much faster than reading from main memory. The benefits of using cache to store data are well known, and to further refine the cache usage to include a partial results cache provides a

Art Unit: 2127

means of guaranteeing that the data being returned makes optimal use of the size limitations of the cache.

56. As per claim 12, Hanson teaches the invention as claimed, including the method as claimed in claim 11, further comprising:

the third thread returning a portion of the result to the client computer (col. 6 lines 9-54).

57. Huff teaches the invention as claimed, including if a second request is received for an additional portion of the result that is stored in the partial results cache, a fourth thread creating a second work item by invoking another receive handler, wherein the second work item is processed by a fifth thread, which invokes another reply handler (col. 4 lines 40-58; col. 6 lines 12-55; col. 7 lines 17-38).

58. Claims 13, 63-64, and 70-71 are rejected under 35 U.S.C. 103(a) as being unpatentable over Huff in view of Hanson as applied to claims 1, 58, and 65 above respectively, and further in view of Ramakrishnan.

59. As per claim 13, Hanson teaches the invention as claimed, including the method as claimed in claim 1, further comprising:

the first thread, the second thread or the third thread indicating that the first thread, the second thread or the third thread has completed a work item (col. 6 lines 9-54).

60. Ramakrishnan teaches the invention as claimed, including the following limitations not shown by the modified Huff:

if a quantum has not expired for the first thread, the second thread or the third thread, then the first thread, the second thread or the third thread being given an additional work item to perform without relinquishing the central processing unit upon which the first thread, the second thread or the third thread was running (col. 9 line 52 - col. 10 line 3).

61. It would have been obvious to one of ordinary skill in the art to combine the modified Huff with Ramakrishnan since the controlling of the amount of time a thread is scheduled to run eliminates livelock and starvation conditions by guaranteeing that a thread services a minimum amount of work and yields after a specified quantum, respectively. Ramakrishnan provides a minimum execution time for a thread, which is specified in terms of work units, such that a thread executes for a minimum amount of time, thereby allowing a thread to perform its requisite task without being interrupted. Additionally, Ramakrishnan provides the benefit of a maximum execution time, also specified in terms of work units, that ensures that one thread does not occupy too many system resources by guaranteeing that it yields after a specified period of time.

62. As per claim 63, Ramakrishnan teaches the invention as claimed, including the computer system as claimed in claim 58, wherein the application server further performs functions of:

monitoring a quantity of work being performed by the computer system (col. 8 lines 13-39; col. 11 line 52 - col. 12 line 8);

determining whether the quantity has exceeded an upper limit (col. 8 lines 13-39; col. 11 line 52 - col. 12 line 8); and

if the quantity has exceeded the upper limit but has not dropped below a lower limit, not accepting new requests into the computer system (col. 8 lines 13-39; col. 11 line 52 - col. 12 line 8).

63. As per claim 64, Ramakrishnan teaches the invention as claimed, including the computer system as claimed in claim 58, wherein the application server further performs functions of:

monitoring an amount of time to return a result by the computer system (col. 8 lines 13-39);

determining whether the amount of time has exceeded an upper limit (col. 8 lines 13-39); and

if the amount of time has exceeded the upper limit but has not dropped below a lower limit, not processing new work items (col. 8 lines 13-39).

64. As per claim 70, Ramakrishnan teaches the invention as claimed, including the computer-readable medium as claimed in claim 65, wherein the method further comprises:

monitoring a quantity of work being performed by the computer system (col. 8 lines 13-39; col. 11 line 52 - col. 12 line 8);

determining whether the quantity has exceeded an upper limit (col. 8 lines 13-39; col. 11 line 52 - col. 12 line 8); and

if the quantity has exceeded the upper limit but has not dropped below a lower limit, not accepting new requests into the computer system (col. 8 lines 13-39; col. 11 line 52 - col. 12 line 8).

65. As per claim 71, Ramakrishnan teaches the invention as claimed, including the computer-readable medium as claimed in claim 65, wherein the method further comprises:

monitoring an amount of time to return a result by the computer system (col. 8 lines 13-39);

determining whether the amount of time has exceeded an upper limit (col. 8 lines 13-39);
and

if the amount of time has exceeded the upper limit but has not dropped below a lower limit, not processing new work items (col. 8 lines 13-39).

66. **Claims 14-16, 18-21, 25 are rejected under 35 U.S.C. 103(a) as being unpatentable over Huff in view of Schaefer et al. (USPN 6,157,927) (hereinafter Schaefer).**

67. As per claim 14, Huff teaches the invention as claimed, including a method in a computer system for servicing requests from multiple client computers, the method comprising:

receiving a request from a client computer to perform a multi-state function (col. 4 lines 40-58; col. 6 lines 12-55);

performing a first task, by a first work handler invoked by a first thread in a ready state, wherein the first task is associated with a first state of the multi-state function (col. 4 lines 40-58; col. 6 lines 12-55);

placing the first thread back in the ready state (col. 6 lines 12-55); and

Art Unit: 2127

performing a second task, by a second work handler invoked by a second thread in the ready state, wherein the second task is associated with a second state of the multi-state function, and the second task performs an operation on the data (col. 6 lines 12-55; col. 7 lines 17-38).

68. Schaefer teaches the invention as claimed, including the following limitations not shown by Huff:

performing the first task includes issuing an asynchronous request for data (col. 13 line 48 - col. 14 line 10); and

receiving the data specified in the asynchronous request (col. 13 line 48 - col. 14 line 10).

69. It would have been obvious to one of ordinary skill in the art to combine Huff and Schaefer since in the case where multiple threads are concurrently accessing a resource, there may be conditions wherein one thread requires data that another thread is producing. In such an instance, a synchronous request would cause the requesting thread to block, since the other thread would not be able to access the resource. By issuing an asynchronous request, the requesting thread may yield, allowing the other thread to perform its processing and return the requested data.

70. As per claim 15, Huff teaches the invention as claimed, including the method as claimed in claim 14, further comprising:

processing the request by a receive handler invoked by a third thread (col. 4 lines 40-58; col. 6 lines 12-55);

creating a work item that specifies the first task (col. 6 lines 12-55; col. 7 lines 17-38);
and

Art Unit: 2127

placing the work item in a work queue that is accessible to the first thread (col. 6 lines 12-45).

71. As per claim 16, Huff teaches the invention as claimed, including the method as claimed in claim 15, wherein the first thread, the second thread, and the third thread are all identical generic threads within a pool of generic threads (col. 4 lines 9-58).

72. As per claim 18, Schaefer teaches the invention as claimed, including the method as claimed in claim 14, wherein the asynchronous request is a request that would otherwise cause the first thread to block (col. 13 line 48 - col. 14 line 10).

73. As per claim 19, Schaefer teaches the invention as claimed, including the method as claimed in claim 18, wherein the asynchronous request is a request that would otherwise cause a blocking condition to occur (col. 13 line 48 - col. 14 line 10).

74. As per claim 20, Schaefer teaches the invention as claimed, including the method as claimed in claim 14, wherein issuing the asynchronous request comprises issuing the asynchronous request to a database manager (col. 2 line 37 - col. 3 line 10).

75. As per claim 21, Schaefer teaches the invention as claimed, including the method as claimed in claim 20, further comprising:

Art Unit: 2127

the database manager placing the asynchronous request on a pending queue (col. 13 line 48 - col. 14 line 10).

76. Huff teaches the invention as claimed, including:

when the data is received, placing a work item associated with the second state on a work queue (col. 6 lines 12-55; col. 7 lines 17-38).

77. As per claim 25, Huff teaches the invention as claimed, including the method as claimed in claim 14, further comprising:

performing additional tasks associated with subsequent function states by additional work handlers invoked by one or more additional threads (col. 4 lines 40-58; col. 6 lines 12-55; col. 7 lines 17-38);

at least some of the additional work handlers issuing additional requests (col. 4 lines 40-58; col. 6 lines 12-55; col. 7 lines 17-38); and

placing threads associated with the at least some of the additional work handlers back in the ready state after issuing the additional requests (col. 4 lines 40-58; col. 6 lines 12-55; col. 7 lines 17-38).

78. Schaefer teaches the invention as claimed, including:

the requests being asynchronous requests (col. 13 line 48 - col. 14 line 10).

79. Claims 17 and 22 are rejected under 35 U.S.C. 103(a) as being unpatentable over Huff in view of Schaefer as applied to claims 15 and 21 above respectively, and further in view of Wight et al. (USPN 6,219,353) (hereinafter Wight).

80. As per claim 17, Wight teaches the invention as claimed, including the method as claimed in claim 15, wherein placing the work item in the work queue comprises placing the work item in a low priority work queue, the method further comprising:

placing a second work item that specifies the second task on a high priority work queue (col. 4 lines 43-59);

a work handler looking for a work item first on the high priority work queue (col. 4 line 60 - col. 5 line 3); and

if no work item exists on the high priority work queue, the work handler looking for the work item on the low priority queue (col. 4 line 60 - col. 5 line 3).

81. It would have been obvious to one of ordinary skill in the art to combine the modified Huff with Wight since the use of priority queues allows the system to ensure that important requests, such as requests with hard deadlines are serviced first. For instance, if a high priority important request is received, it may be detrimental to simply place the request in a first-in-first-out queue. A deadline may be missed or the system may lag. By providing separate queues for high priority and low priority requests, it can be ensured that the most important requests are serviced first.

82. As per claim 22, Wight teaches the invention as claimed, including the method as claimed in claim 21, wherein the work queue is a high priority work queue, the method further comprising:

Art Unit: 2127

placing a first work item that specifies the first task on a low priority work queue (col. 4 lines 43-59);

a work handler looking for a work item first on the high priority work queue (col. 4 line 60 - col. 5 line 3); and

if no work item exists on the high priority work queue, the work handler looking for the work item on the low priority work queue (col. 4 line 60 - col. 5 line 3).

83. Claims 23-24 are rejected under 35 U.S.C. 103(a) as being unpatentable over Huff in view of Schafer as applied to claim 14 above, and further in view of Hanson.

84. As per claim 23, Hanson teaches the invention as claimed, including the method as claimed in claim 14, further comprising returning a result of the first task and the second task by invoking third thread, which in turn invokes a reply handler to return the result to the client computer (col. 6 lines 9-54).

85. It would have been obvious to one of ordinary skill in the art to combine the modified Huff with Hanson since the thread allocation procedure taught by Huff deals only with allocating an incoming request to a thread for processing, but fails to specifically state how the processing results are returned to the client. Hanson makes up for this deficiency with a method of invoking a reply thread in a distributed processing system that allows the agent or server to handle all requests transparently from the client. This simplifies the processing and allows system threads to be allocated in an efficient manner.

Art Unit: 2127

86. As per claim 24, Huff teaches the invention as claimed, including the method as claimed in claim 23, wherein the first thread, the second thread, and the third thread are all identical generic threads within a pool of generic threads (col. 4 lines 9-58).

87. **Claims 26-27 and 31-32 are rejected under 35 U.S.C. 103(a) as being unpatentable over Huff in view of Wight.**

88. As per claim 26, Huff teaches the invention as claimed, including a method in a computer system for servicing requests from multiple client computers, the method comprising:

determining that work is available after receiving a request from a client computer (col. 4 lines 40-58; col. 6 lines 12-55);

89. Wight teaches the invention as claimed, including the following limitations not shown by Huff:

when work is available, a first work handler invoked by a first thread looking in a first work queue for a first work item corresponding to the work (col. 4 line 60 - col. 5 line 3); and

if the first work item is not found in the first work queue, the first work handler looking in a second work queue for the first work item (col. 4 line 60 - col. 5 line 3).

90. It would have been obvious to one of ordinary skill in the art to combine Huff and Wight since the use of priority queues allows the system to ensure that important requests, such as requests with hard deadlines are serviced first. For instance, if a high priority important request is received, it may be detrimental to simply place the request in a first-in-first-out queue. A

Art Unit: 2127

deadline may be missed or the system may lag. By providing separate queues for high priority and low priority requests, it can be ensured that the most important requests are serviced first.

91. As per claim 27, Huff teaches the invention as claimed, including the method as claimed in claim 26, further comprising:

receiving a request from a client computer to perform a task (col. 4 lines 40-58; col. 6 lines 12-55);

creating the first work item that specifies the task (col. 4 lines 40-58; col. 6 lines 12-55);

placing the first work item in the second queue (col. 6 lines 12-55; col. 7 lines 17-38);

and

indicating that the work is available (col. 6 lines 12-55).

92. As per claim 31, Wight teaches the invention as claimed, including the method as claimed in claim 26, wherein the computer system includes multiple work queues, including the first work queue and the second work queue, each of the multiple work queues are associated with a priority level, and wherein the method further comprises:

looking for the first work item first in a work queue associated with a highest priority level (col. 4 line 60 - col. 5 line 3); and

if the first work item is not found in the work queue associated with the highest priority level, looking for the first work item in each of the multiple work queues in descending priority order until the first work item is found (col. 4 line 60 - col. 5 line 3).

Art Unit: 2127

93. As per claim 32, Huff teaches the invention as claimed, including the method as claimed in claim 31, wherein the request from the client computer is a request to perform a function having multiple states, wherein each state of the multi-state function is performed by a work handler invoked by a subsequent thread based on work items placed in the multiple work queues (col. 4 lines 40-58; col. 6 lines 12-55).

94. Wight teaches the invention as claimed, including the work items are placed in higher and higher priority work queues as execution of the multi-state function progresses through the multiple states (col. 4 line 60 - col. 5 line 3).

95. **Claims 28-30 are rejected under 35 U.S.C. 103(a) as being unpatentable over Huff in view of Wight as applied to claim 26 above, and further in view of Schaefer.**

96. As per claim 28, Wight teaches the invention as claimed, including the method as claimed in claim 26, further comprising:

the first work handler performing a task specified in the first work item (col. 4 line 60 - col. 5 line 3).

97. Schaefer teaches the invention as claimed, including the following limitations not shown by the modified Huff:

issuing an asynchronous request for data (col. 13 line 48 - col. 14 line 10).

98. It would have been obvious to one of ordinary skill in the art to combine the modified Huff with Schaefer since in the case where multiple threads are concurrently accessing a resource, there may be conditions wherein one thread requires data that another thread is

Art Unit: 2127

producing. In such an instance, a synchronous request would cause the requesting thread to block, since the other thread would not be able to access the resource. By issuing an asynchronous request, the requesting thread may yield, allowing the other thread to perform its processing and return the requested data.

99. As per claim 29, Huff teaches the invention as claimed, including the method as claimed in claim 28, further comprising:

receiving the data (col. 4 lines 40-58; col. 6 lines 12-55);

placing a second work item on the first work queue (col. 4 lines 40-58; col. 6 lines 12-55); and

indicating that additional work is available (col. 4 lines 40-58; col. 6 lines 12-55).

100. As per claim 30, Wight teaches the invention as claimed, including the method as claimed in claim 29, further comprising:

a second work handler invoked by a second thread looking in the first work queue for the second work item when the second work is available (col. 4 line 60 - col. 5 line 3); and

performing a second task specified in the second work item (col. 4 line 60 - col. 5 line 3).

101. **Claim 33 is rejected under 35 U.S.C. 103(a) as being unpatentable over Huff in view of Challenger et al. (USPN 6,026,413) (hereinafter Challenger).**

Art Unit: 2127

102. As per claim 33, Huff teaches the invention as claimed, including a method in a computer system for servicing requests from multiple client computers, the method comprising:

receiving, from a client computer, a request to perform a first task (col. 4 lines 40-58; col. 6 lines 12-55).

103. Challenger teaches the invention as claimed, including the following limitations not shown by Huff:

evaluating the first task, by a first handler invoked by a first thread, to determine whether the first task includes complex or long-running logic (col. 30 line 34 - col. 33 line 50); and

if the first task includes complex or long-running logic, performing the first task by a second handler invoked by a second thread (col. 30 line 34 - col. 33 line 50).

104. It would have been obvious to one of ordinary skill in the art to combine Huff and Challenger since a complex or long-running request may require special consideration. Specifically, a request that requires more processing than is required for a typical task may need to be serviced at a higher priority level since it may take longer to complete. Additionally, the amount of time required to process the thread might be longer than the specified scheduling epoch, thereby requiring several time slices to complete the processing. A number of additional concerns may arise, requiring further attention. Therefore, it is important to identify these types of tasks and take the appropriate action.

105. **Claims 34-35 are rejected under 35 U.S.C. 103(a) as being unpatentable over Huff in view of Challenger as applied to claim 34 above, and further in view of Wight.**

Art Unit: 2127

106. As per claim 34, Wight teaches the invention as claimed, including the method as claimed in claim 33, wherein the first thread is a thread within a first group of threads having a first priority level, and the second thread is a thread within a second group of threads having a second priority level that is lower than the first priority level (col. 4 line 43 - col. 5 line 3).

107. It would have been obvious to one of ordinary skill in the art to combine the modified Huff and Wight since the use of priority queues allows the system to ensure that important requests, such as requests with hard deadlines are serviced first. For instance, if a high priority important request is received, it may be detrimental to simply place the request in a first-in-first-out queue. A deadline may be missed or the system may lag. By providing separate queues for high priority and low priority requests, it can be ensured that the most important requests are serviced first.

108. As per claim 35, Huff teaches the invention as claimed, including the method as claimed in claim 34, further comprising:

receiving a request to perform a second task (col. 4 lines 40-58; col. 6 lines 12-55).

109. Challenger teaches the invention as claimed, including evaluating the second task to determine whether the second task includes complex or long-running logic (col. 30 line 34 - col. 33 line 50); and

if the second task does not include complex or long-running logic and a processor is not available for performing the second task, preempting the second thread and performing the second task by a third thread in the first group of threads (col. 30 line 34 - col. 33 line 50).

Art Unit: 2127

110. Claim 36 is rejected under 35 U.S.C. 103(a) as being unpatentable over Huff in view of Challenger as applied to claim 33 above, and further in view of Hanson.

111. As per claim 36, Hanson teaches the invention as claimed, including the method as claimed in claim 33, further comprising:

the second handler returning a result of the first task (col. 6 lines 9-54); and

using the result, performing a second task by a third handler invoked by a third thread of the first group of threads (col. 6 lines 9-54).

112. It would have been obvious to one of ordinary skill in the art to combine the modified Huff and Hanson since the thread allocation procedure taught by Huff deals only with allocating an incoming request to a thread for processing, but fails to specifically state how the processing results are returned to the client. Hanson makes up for this deficiency with a method of invoking a reply thread in a distributed processing system that allows the agent or server to handle all requests transparently from the client. This simplifies the processing and allows system threads to be allocated in an efficient manner.

113. Claim 37 is rejected under 35 U.S.C. 103(a) as being unpatentable over Huff in view of Challenger in view of Hanson as applied to claim 36 above, and further in view of Wight.

114. As per claim 37, Wight teaches the invention as claimed, including the method as claimed in claim 36, wherein the first task is specified in a first work item and the second task is specified in a second work item, the method further comprising:

the first handler obtaining the first work item from a first work queue (col. 4 line 60 - col. 5 line 3); and

the third handler obtaining the second work item from a second work queue (col. 4 line 60 - col. 5 line 3).

115. It would have been obvious to one of ordinary skill in the art to combine the modified Huff and Wight since the use of priority queues allows the system to ensure that important requests, such as requests with hard deadlines are serviced first. For instance, if a high priority important request is received, it may be detrimental to simply place the request in a first-in-first-out queue. A deadline may be missed or the system may lag. By providing separate queues for high priority and low priority requests, it can be ensured that the most important requests are serviced first.

116. Claims 53-54, 60, and 67 are rejected under 35 U.S.C. 103(a) as being unpatentable over Huff in view of Hanson as applied to claims 49, 58, and 65 above respectively, and further in view of Schaefer.

117. As per claim 53, Huff teaches the invention as claimed, including the application program as claimed in claim 49, wherein the thread can invoke multiple work handlers, where some of the multiple work handlers are designed to perform tasks associated with various states of a multi-state function (col. 4 lines 40-58; col. 6 lines 12-55), wherein the thread is then placed back in a ready state to execute a subsequent work item (col. 6 lines 12-55).

Art Unit: 2127

118. Schaefer teaches the invention as claimed, including the following limitations not shown by the modified Huff:

at least some of the multiple work handlers issue asynchronous requests for data when a state transition is to be performed (col. 13 line 48 - col. 14 line 10).

119. It would have been obvious to one of ordinary skill in the art to combine the modified Huff with Hanson since the thread allocation procedure taught by Huff deals only with allocating an incoming request to a thread for processing, but fails to specifically state how the processing results are returned to the client. Hanson makes up for this deficiency with a method of invoking a reply thread in a distributed processing system that allows the agent or server to handle all requests transparently from the client. This simplifies the processing and allows system threads to be allocated in an efficient manner.

120. As per claim 54, Hanson teaches the invention as claimed, including the application program as claimed in claim 53, wherein when the data is returned, a second work handler is executed (col. 6 lines 9-54).

121. As per claim 60, Huff teaches the invention as claimed, including the computer system as claimed in claim 58, wherein the application server further performs functions of:

receiving a request from a client computer to perform a multi-state function (col. 4 lines 40-58; col. 6 lines 12-55);

Art Unit: 2127

a first thread within the pool of threads performing a first task by invoking a first work handler, wherein the first task is associated with a first state of the multi-state function (col. 4 lines 40-58; col. 6 lines 12-55);

placing the first thread back in a ready state (col. 6 lines 12-55); and

a second thread within the pool of threads performing a second task by invoking a second work handler, wherein the second, task is associated with a second state of the multi-state function, and the second task performs an operation on the data (col. 6 lines 12-55; col. 7 lines 17-38).

122. Schaefer teaches the invention as claimed, including the following limitations not shown by the modified Huff:

performing the first task includes issuing an asynchronous request for data (col. 13 line 48 - col. 14 line 10);

receiving the data specified in the asynchronous request (col. 13 line 48 - col. 14 line 10).

123. As per claim 67, Huff teaches the invention as claimed, including the computer-readable medium as claimed in claim 65, wherein the method further comprises:

receiving a request from a client computer to perform a multi-state function (col. 4 lines 40-58; col. 6 lines 12-55);

a first thread within the pool of threads performing a first task by invoking a first work handler, wherein the first task is associated with a first state of the multi-state function (col. 4 lines 40-58; col. 6 lines 12-55);

placing the first thread back in a ready state (col. 6 lines 12-55); and

a second thread within the pool of threads performing a second task by invoking a second work handler, wherein the second task is associated with a second state of the multi-state function, and the second task performs an operation on the data (col. 6 lines 12-55; col. 7 lines 17-38).

124. Schaefer teaches the invention as claimed, including the following limitations not shown by the modified Huff:

performing the first task includes issuing an asynchronous request for data (col. 13 line 48 - col. 14 line 10);

receiving the data specified in the asynchronous request (col. 13 line 48 - col. 14 line 10).

125. Claims 57, 62, and 69 are rejected under 35 U.S.C. 103(a) as being unpatentable over Huff in view of Hanson as applied to claims 49, 58, and 65 above respectively, and further in view of Challenger.

126. As per claim 57, Challenger teaches the invention as claimed, including the application program as claimed in claim 49, further comprising one or more complex logic handlers that can be invoked by a second type of thread, wherein a thread of the second type is executed when a request from a client computer involves execution of complex or long-running logic (col. 30 line 34 - col. 33 line 50).

127. It would have been obvious to one of ordinary skill in the art to combine the modified Huff with Challenger since a complex or long-running request may require special consideration. Specifically, a request that requires more processing than is required for a typical task may need

Art Unit: 2127

to be serviced at a higher priority level since it may take longer to complete. Additionally, the amount of time required to process the thread might be longer than the specified scheduling epoch, thereby requiring several time slices to complete the processing. A number of additional concerns may arise, requiring further attention. Therefore, it is important to identify these types of tasks and take the appropriate action.

128. As per claim 62, Huff teaches the invention as claimed, including the computer system as claimed in claim 58, wherein the application server further performs functions of:

receiving, from a client computer, a request to perform a first task (col. 4 lines 40-58; col. 6 lines 12-55).

129. Challenger teaches the invention as claimed, including the following limitations not shown by the modified Huff:

evaluating the first task, by a first handler invoked by a first thread within the pool of threads, to determine whether the first task includes complex or long-running logic (col. 30 line 34 - col. 33 line 50); and

if the first task includes complex or long-running logic, performing the first task by a second handler invoked by a second thread that is not within the pool of threads (col. 30 line 34 - col. 33 line 50).

130. As per claim 69, Huff teaches the invention as claimed, including the computer-readable medium as claimed in claim 65, wherein the method further comprises:

receiving, from a client computer, a request to perform a first task (col. 4 lines 40-58; col. 6 lines 12-55).

131. Challenger teaches the invention as claimed, including the following limitations not shown by the modified Huff:

evaluating the first task, by a first handler invoked by a first thread within the pool of threads, to determine whether the first task includes complex or long-running logic (col. 30 line 34 - col. 33 line 50); and

if the first task includes complex or long-running logic, performing the first task by a second handler invoked by a second thread that is not within the pool of threads (col. 30 line 34 - col. 33 line 50).

132. Claims 61 and 68 rejected under 35 U.S.C. 103(a) as being unpatentable over Huff in view of Hanson as applied to claims 58 and 65 above respectively, and further in view of Wight.

133. As per claim 61, Wight teaches the invention as claimed, including the computer system as claimed in claim 58, wherein the application server further performs functions of:

determining that work is available after receiving a request from a client computer (col. 4 line 60 - col. 5 line 3);

when work is available, a first thread within the pool of threads invoking a first work handler to look in a first work queue for a first work item corresponding to the work (col. 4 line 60 - col. 5 line 3); and

Art Unit: 2127

if the first work item is not found in the first work queue, the first work handler looking in a second work queue for the first work item (col. 4 line 60 - col. 5 line 3).

134. It would have been obvious to one of ordinary skill in the art to combine the modified Huff with Wight since the use of priority queues allows the system to ensure that important requests, such as requests with hard deadlines are serviced first. For instance, if a high priority important request is received, it may be detrimental to simply place the request in a first-in-first-out queue. A deadline may be missed or the system may lag. By providing separate queues for high priority and low priority requests, it can be ensured that the most important requests are serviced first.

135. As per claim 68, Wight teaches the invention as claimed, including the computer-readable medium as claimed in claim 65, wherein the method further comprises:

determining that work is available after receiving a request from a client computer (col. 4 line 60 - col. 5 line 3);

when work is available, a first work handler invoked by a first thread within the pool of threads looking in a first work queue for a first work item corresponding to the work (col. 4 line 60 - col. 5 line 3); and

if the first work item is not found in the first work queue, the first work handler looking in a second work queue for the first work item (col. 4 line 60 - col. 5 line 3).

Art Unit: 2127

Conclusion

136. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Syed J Ali whose telephone number is (703) 305-8106. The examiner can normally be reached on Mon-Fri 8-5:30, 2nd Friday off.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Meng-Ai T An can be reached on (703) 305-9678. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).



Syed Ali
April 23, 2004



MENG-AI T. AN
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100